

**PATENTS**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

APPELLANTS:	Bergman, et al.	Examiner: Dao, Thuy Chan
Serial No.:	10/630,959	Group: Art Unit 2192
Filed:	July 30, 2003	Docket: YOR920030056US1 (8728-607)
For:	<b>SYSTEMS AND METHODS FOR GENERATING AND DISTRIBUTING EXECUTABLE PROCEDURES FOR TECHNICAL DESK-SIDE SUPPORT</b>	

**Mail Stop Appeal Brief-Patents**  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

In response to the Final Office Action dated May 8, 2007 finally rejecting 1-34 and the Notice of Panel Decision made October 1, 2007, Applicants appeal pursuant to the Notice of Appeal filed on August 8, 2007 and submit this appeal brief.

## **TABLE OF CONTENTS**

	<b><u>Page</u></b>
1. REAL PARTY IN INTEREST	1
2. RELATED APPEALS AND INTERFERENCES	1
3. STATUS OF CLAIMS	1
4. STATUS OF AMENDMENTS	1
5. SUMMARY OF CLAIMED SUBJECT MATTER	2
6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL	3
7. ARGUMENT	4
A.    The Claim Rejections Under 35 U.S.C. §102 Are Legally Deficient	4
i.    Claim 20	4
ii.   Claim 26	6
iii.  Claim 30	9
iv.   Claims 27-29	10
B.    The Claim Rejections Under 35 U.S.C. §103 Are Legally Deficient	11
i.    Claim 30	11
ii.   Claim 20	13
iii.  Claims 21-25 and 31-34	15
C.    Conclusion	16
8. CLAIMS APPENDIX	17
9. EVIDENCE APPENDIX	21
10. RELATED PROCEEDINGS APPENDIX	22

**1. Real Party in Interest**

The real party in interest is International Business Machines Corporation, the assignee of the entire right, title, and interest in and to the subject application by virtue of an assignment of record.

**2. Related Appeals and Interferences**

(None)

**3. Status of Claims**

Claims 20-34, are pending, stand rejected, and are under appeal.

A copy of the Claims as pending is presented in the Claims Appendix.

**4. Status of Amendments**

By the Amendment dated August 8, 2007, Claims 1-19 were cancelled without prejudice. This amendment was entered.

## **5. Summary of Claimed Subject Matter**

The present invention relates to systems and methods for creating procedures for providing technical support and executing the same.

Referring to Claim 20; A system for providing technical support includes a client device comprising an application for monitoring and recording a procedure that is performed using said client device and generating an execution trace representing an instance of said procedure (see for example, page 12, lines 5-17); a procedure trace repository for storing execution traces (see for example, page 12, line 18-19); and a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure, wherein said procedure can be automatically performed on the client by invoking the reusable executable procedure (see for example, page 12, line 21 to page 13, line 9).

Referring to Claim 26; A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for generating a reusable executable procedure is claimed (see for example, page 7, line 9-24). The method steps including obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure (see for example, page 8, lines 6-12); and processing said execution traces to create a reusable executable procedure associated with said procedure, wherein said procedure can be automatically performed by invoking the reusable executable procedure (see for example, page 9, lines 5-22).

Referring to Claim 30; A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for executing a reusable procedure is claimed (see for example, page 7, line 9-24 and page 15, lines 16-18). The method steps including launching a reusable executable procedure (see for example,

page 15, line 20-24); automatically executing procedure steps associated with said reusable executable procedure (see for example, page 16, line 4 to page 17, line 5); and relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable executable procedure cannot be successfully executed (see for example, page 17, lines 14-29).

**6. Grounds of Rejection to be Reviewed on Appeal**

**A.** Claims 20, and 26 are rejected under 35 U.S.C. 102(e) as being anticipated by Messinger (US Patent No. 7,000,187).

**B.** Claims 20, and 26 are rejected under 35 U.S.C. 102(e) as being anticipated by Bala (US Patent Application No. 2004/0130572).

**C.** Claims 26-29 are rejected under 35 U.S.C. 102(b) as being anticipated by Horvitz (US Patent No. 6,260,035).

**D.** Claim 30 are rejected under 35 U.S.C. 102(b) as being anticipated by Mayuzumi (US Patent No. 6,134,644).

**E.** Claims 30-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Horvitz in view of Harel (U.S. Patent No 6,384,843)

**F.** Claims 20-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Horvitz in view of Sullivan (U.S. Patent No. 6,615,240).

7. **Argument**

A. **The Claim Rejections Under 35 U.S.C. 102 Are Legally Deficient.**

Under 35 U.S.C. §102, a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. The identical invention must be shown in as complete detail as is contained in the claim. See MPEP §2131.

i. **Claims 20**

Referring to Claim 20; Claim 20 stands rejected as being anticipated, separately, by Messinger (US Patent No. 7,000,187) and Bala (US Patent Application No. 2004/0130572).

Claim 20 claims:

*A system for providing technical support, comprising:*

*a client device comprising an application for monitoring and recording a procedure that is performed using said client device and generating an execution trace representing an instance of said procedure;*

*a procedure trace repository for storing execution traces; and*

*a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure, wherein said procedure can be automatically performed on the client by invoking the reusable executable procedure.*

It has not demonstrated how the cited references Messinger or Bala teach “a server for processing a plurality of execution traces associated with instances of an executed procedure to

generate a reusable executable procedure, wherein said procedure can be automatically performed on the client by invoking the reusable executable procedure” as claimed in Claim 20.

With regard to Messinger; Messinger teaches a method for displaying a sequence of instructions associated with a task in a graphical overlay, for example, for software training (see Abstract). Messinger does not teach “a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure.” Messinger merely teaches how an “on-line coach” operates based on user (trainee) input (see col. 2, lines 4-8). An “on-line coach” is not analogous to “a server that processes a plurality of execution traces associated with instances of an executed procedure to generate a reusable procedure,” as claimed in Claim 20. For example, nowhere does Messinger teach or suggest how an “on-line coach” could be generated, much less “processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure,” as claimed in Claim 20. Therefore, Messinger fails to teach all the limitations of Claim 20.

With regard to Bala; Bala teaches methods for authoring and executing wizards, wherein wizards are updated through a feedback system (see Abstract). Bala fails teach “a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure” as claimed in Claim 20. Bala merely teaches how a user can create script for a task (see paragraphs [0074-000079]). Bala is totally devoid of description related multiple scripts, much less “processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure,” as claimed in Claim 20. Therefore, Bala fails to teach all the limitations of Claim 20.

Accordingly, the rejections of Claim 20 should be overruled.

**ii. Claims 26**

Referring to Claim 26; Claim 26 stands rejected as being anticipated, separately, by Messinger, Bala and Horvitz (US Patent No. 6,260,035).

Claim 26 claims:

*A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for generating a reusable executable procedure, the method steps comprising:*

*obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure; and*

*processing said execution traces to create a reusable executable procedure associated with said procedure, wherein said procedure can be automatically performed by invoking the reusable executable procedure.*

With regard to Messinger; Messinger teaches a method for displaying a sequence of instructions associated with a task in a graphical overlay, for example, for software training (see Abstract). Messinger does not teach “obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure; and processing said execution traces to create a reusable executable procedure associated with said procedure” as claimed in Claim 26. Although Messinger arguably teaches a process of recording a new task sequence (in



block 385 of FIG. 8), even assuming that the process of recording a new task sequence is an execution trace that represents an execution instance of a procedure, Messinger fails to teach “obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure; and processing said execution traces to create a reusable executable procedure associated with said procedure” - for example, Messinger merely adds a new task to a task list. Each task of Messinger is individually maintained and there is no processing to create a reusable executable procedure, essentially as claimed. Therefore, Messinger fails to teach all the limitations of Claim 26.

With regard to Bala; Bala teaches methods for authoring and executing wizards, wherein wizards are updated through a feedback system (see Abstract). Bala does not teach the process of “obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure; and processing said execution traces to create a reusable executable procedure associated with said procedure” as claimed in Claim 26. Bala merely teaches how a user can create script for a task (see paragraphs [0074-000079]). Bala is totally devoid of description related multiple scripts, much less “obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure; and processing said execution traces to create a reusable executable procedure associated with said procedure,” as claimed in Claim 26. Therefore, Bala fails to teach all the limitations of Claim 26.

With regard to Horvitz; Horvitz teaches a method for an intelligent user interface system including a reasoning model to compute a probability of user’s intentions (see Abstract). Horvitz does not teach methods for “obtaining a plurality of execution traces, wherein each execution

trace represents an execution instance of a procedure; and processing said execution traces to create a reusable executable procedure associated with said procedure” as claimed in Claim 26. In formulating the rejection, the Examiner relies on Horvitz’s teachings (in FIG. 2, blocks 56-60, Col 7, lines 29-51, FIG. 8, blocks 112-118, Col. 14, lines 60-67) as teaching the claimed process of “obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure.” Horvitz teaches a process of monitoring and recording user atomic events when interacting with an application, for the purpose of providing intelligent user assistance. Horvitz teaches methods for observing and modeling user interaction behaviors to create models that enable the system to determine when a user experiences difficulties in using an application. The models are processed using inference engines to form and evaluate multiple hypotheses of what assistance a user may need when interacting with an application (see, e.g., Col. 6, lines 4-29; Col. 7, lines 29-67). The modeled event database (106) in FIG. 8 of Horvitz relates to a modeled event database definition (see Col. 15, lines 1-5, and Col 13, lines 30-65), but the “modeled events” as taught by Horvitz are not fairly characterized as being “reusable executable procedures” within the context of Claim 26. For example, the modeled events correspond to difficulties in using software (col. 8, lines 34-48). Thus, the modeled events of Horvitz are the antithesis of an “execution trace represents an execution instance of a procedure” - the modeled events evidence a procedure breakdown. Further, modeled events of Horvitz are atomic (see col. 3, lines 36-37) - atomic events are clearly not an execution trace representing an execution instance of a procedure. Even assuming that modeled events are procedures, it is clear that it would not be desirable to repeat these modeled events (difficulties (see col. 8, lines 34-48)), thus, Horvitz does not teach a “procedure can be automatically performed by invoking the reusable executable procedure.” Therefore, Horvitz fails to teach all the limitations of Claim 26.

Accordingly, the rejections of Claim 26 should be overruled.

**iii. Claims 30**

Referring to Claim 30; Claim 30 stands rejected as being anticipated by Mayuzumi (US Patent No. 6,134,644).

Claim 30 claims:

*A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for executing a reusable procedure, the method steps comprising:*  
*launching a reusable executable procedure;*  
*automatically executing procedure steps associated with said reusable executable procedure; and*  
*relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable executable procedure cannot be successfully executed.*

The Examiner's anticipation rejection of Claim 30 is based on a mischaracterization and misunderstanding of the teachings of Mayuzumi as applied to claim 30. There is no reasonable basis, whatsoever, as construing Mayuzumi (FIG. 16 and supporting explanation in Col 20, lines 15-29 and Col. 21, lines 1-17) as teaching the claimed process of "launching a reusable executable procedure; automatically executing procedure steps associated with said reusable

executable procedure; and relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable executable procedure cannot be successfully executed.”

Mayuzumi does not specifically teach a “reusable executable procedure” within the context of the claimed inventions. Mayuzumi teaches a method for dynamically generating a sequence of data parts based on the type of help information needed at a given time (i.e., depending on a given event or combination of events) (see, e.g., Col. 19, lines 35 – Col. 20, line 9). In this regard, the sequence of data parts is not maintained as “reusable executable procedures” per se, as the sequence of data parts is dynamically generated.

Moreover, Mayuzumi teaches in FIG. 16 a method in which help screens are presented to a user illustrating a process flow for sequential work that can be performed to recover from an error (see, Col. 20, lines 52-67, FIG. 15). In this regard, the process flows are not automatically executed as part of a reusable executable procedure, essentially as claimed in Claim 30, but are merely dynamically generated based on one or more events, and then displayed to provide user guidance in manually performing a given task as suggested in the displayed process flow. Therefore, Mayuzumi fails to teach all the limitations of Claim 30.

Accordingly, the rejections of Claim 30 should be overruled.

#### **iv. Claims 27-29**

Claims 26-29 depend from Claim 26. The dependent claims are believed to be allowable for at least the reasons given for Claim 26. Reconsideration of the rejection is respectfully requested.

**B. The Claim Rejections Under 35 U.S.C. 103 Are Legally Deficient.**

In rejecting claims under 35 U.S.C. §103, the Examiner bears the initial burden of presenting a *prima facie* case of obviousness. In re Rijckaert, 9 F.3d 1531, 1532 (Fed. Cir. 1993). The burden of presenting a *prima facie* case of obviousness is only satisfied by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988). A *prima facie* case of obviousness is established when the teachings of the prior art itself would appear to have suggested the claimed subject matter to one of ordinary skill in the art. In re Bell, 991 F.2d 781, 782 (Fed. Cir. 1993). If the Examiner fails to establish a *prima facie* case, the rejection is improper and must be overturned. In re Rijckaert, 9 F.3d at 1532 (citing In re Fine, 837 F.2d at 1074).

**i. Claim 30**

It is respectfully submitted that at the very least, the combined teachings of Horvitz and Harel are legally deficient to establish a *prima facie* case of anticipation against independent Claim 30.

Claim 30 claims:

*A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for executing a reusable procedure, the method steps comprising:*  
*launching a reusable executable procedure;*  
*automatically executing procedure steps associated with said reusable executable procedure; and*

*relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable executable procedure cannot be successfully executed.*

Horvitz teaches a method for an intelligent user interface system including a reasoning model to compute a probability of user's intentions (see Abstract). Horvitz fails to teach or suggest "launching a reusable executable procedure; automatically executing procedure steps associated with said reusable executable procedure; and relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable executable procedure cannot be successfully executed" as claimed in Claim 30. Horvitz teaches a process of monitoring and recording user atomic events when interacting with an application, for the purpose of providing intelligent user assistance. Horvitz teaches methods for observing and modeling user interaction behaviors to create models that enable the system to determine when a user experiences difficulties in using an application. The help of Horvitz is provided in the form of a help dialog box (see Fig. 23). Clearly a help dialog box is not analogous to "reusable executable procedure." For example, the help dialog box provides tutorial information for a user to perform a procedure and as such does not perform "relinquishing control of execution of said reusable procedure to a user" as claimed in Claim 30 (the user dialog box does not take control of execute and therefore cannot perform relinquishing). Therefore, Horvitz fails to teach or suggest all the limitations of Claim 30.

Harel teaches a computerized apparatus for identifying human difficulties (see Abstract). Harel fails to teach or suggest "launching a reusable executable procedure; automatically executing procedure steps associated with said reusable executable procedure; and relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable

executable procedure cannot be successfully executed” as claimed in Claim 30. Harel teaches a system including a human difficulty identifier for identifying difficulties in a software system. A system which merely identifies operations which cause difficulties for a user in a system is not analogous to “reusable executable procedure.” For example, the human difficulty identifier does not perform “relinquishing control of execution of said reusable procedure to a user” as claimed in Claim 30 (similar to the help dialog of Horvitz, human difficulty identifier system monitors a system and does not take control of execution and therefore cannot perform relinquishing). Therefore, Harel fails to cure the deficiencies of Horvitz.

The combined teachings of Horvitz and Harel teach a method for identifying operations which cause a user difficulty and presenting a help dialog. The combined teachings of Horvitz and Harel fail to teach or suggest “launching a reusable executable procedure; automatically executing procedure steps associated with said reusable executable procedure; and relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable executable procedure cannot be successfully executed”, essentially as claimed in Claim 30.

Accordingly, the rejection should be overruled.

**ii. Claim 20**

It is respectfully submitted that at the very least, the combined teachings of Horvitz and Sullivan are legally deficient to establish a prima facie case of anticipation against independent Claim 20.

Claim 20 claims:

*A system for providing technical support, comprising:*

*a client device comprising an application for monitoring and recording a procedure that is performed using said client device and generating an execution trace representing an instance of said procedure;*

*a procedure trace repository for storing execution traces; and*

*a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure, wherein said procedure can be automatically performed on the client by invoking the reusable executable procedure.*

Horvitz teaches a method for an intelligent user interface system including a reasoning model to compute a probability of user's intentions (see Abstract). Horvitz fails to teach or suggest "a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure" as claimed in Claim 20. For example, Horvitz teaches a process of monitoring and recording user atomic events when interacting with an application, for the purpose of providing intelligent user assistance. Horvitz is unrelated to "processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure." Indeed, as noted on page 15 of the Final Office Action, "Horvitz does not explicitly disclose a server [for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure]." Therefore, Horvitz fails to teach or suggest all the limitations of Claim 20.

Sullivan teaches a method for automated technical support in a computer network (see Abstract). Sullivan fails to teach or suggest "a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable



procedure” as claimed in Claim 20. Sullivan teaches a guided self help process (see col. 4, lines 65-66). Sullivan merely teaches how a guided self help system operates. Nowhere does Sullivan teach or suggest how to generated a guided self help system, much less “processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure” Therefore, Sullivan fails to cure the deficiencies of Horvitz.

The combined teachings of Horvitz and Sullivan teach a method for monitoring and recording user events when interacting with an application and providing guided self help. The combined teachings of Horvitz and Sullivan fail to teach or suggest “a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure” essentially as claimed in Claim 20.

Accordingly, the rejection should be overruled.

**iii. Claims 21-25 and 31-34**

Claims 21-25 depend from Claim 20. Claims 31-34 depend from Claim 30. The dependent claims are believed to be allowable for at least the reasons given for Claims 20 and 30. Reconsideration of the rejection is respectfully requested.

### **C. Conclusion**

The claimed invention is not disclosed or suggested by the teachings of the applied prior art references, either alone or in combination. The Examiner has failed to establish a case of anticipation of the presently claimed method under 35 U.S.C. §102 in view of Messinger in view of Claims 20 and 26, and Bala in view of Claims 20 and 26, and Horvitz in view of Claim 26, and further in view of Mayuzumi in view of Claim 30. Moreover, the Examiner has failed to establish a case of obviousness of the presently claimed method under 35 U.S.C. §103 in view of Horvitz and Harel with respect to Claims 30-34 and in view of Horvitz and Sullivan with respect to Claims 20-25 for at least the reasons noted above. Claims 21-25, 27-29 and 31-34 are believed to be allowable for at least the reasons given for the respective independent claims. Accordingly, it is respectfully requested that the Board overrule the rejections of Claims 20-34.

Respectfully Submitted,

Date: November 1, 2007

By: /Nathaniel T. Wallace/  
Nathaniel T. Wallace  
Reg. No. 48,909  
Attorney for Appellants

**F. CHAU & ASSOCIATES, LLP**  
130 Woodbury Road  
Woodbury, New York 11797  
TEL: (516) 692-8888  
FAX: (516) 692-8889

**8. CLAIMS APPENDIX**

20. A system for providing technical support, comprising:

a client device comprising an application for monitoring and recording a procedure that is performed using said client device and generating an execution trace representing an instance of said procedure;

a procedure trace repository for storing execution traces; and

a server for processing a plurality of execution traces associated with instances of an executed procedure to generate a reusable executable procedure, wherein said procedure can be automatically performed on the client by invoking the reusable executable procedure.

.

21. The system of claim 20, further comprising a library for storing reusable executable procedures.

22. The system of claim 20, wherein the server distributes a reusable executable procedure to a client device comprising an execution engine for executing said reusable executable procedure.

23. The system of claim 22, wherein said reusable executable procedure is executed for upgrading software residing on the client device.

24. The system of claim 22, wherein said reusable executable procedure is executed for providing diagnostic support.

25. The system of claim 22, wherein an execution engine of a client device comprises means for allowing a user to manually execute at least a portion of said reusable executable device and generating an execution trace representing said manual execution, wherein said execution trace representing said manual execution is processed by said server to augment said reusable executable procedure.

26. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for generating a reusable executable procedure, the method steps comprising:

obtaining a plurality of execution traces, wherein each execution trace represents an execution instance of a procedure; and

processing said execution traces to create a reusable executable procedure associated with said procedure, wherein said procedure can be automatically performed by invoking the reusable executable procedure.

27. The program storage device of claim 26, wherein the instructions for obtaining an execution trace comprise instructions for monitoring and recording a sequence of actions that are performed by an individual when executing an instance of said procedure.

28. The program storage device of claim 26, wherein the instructions for processing said execution traces comprise instructions for performing the steps of:

aligning said execution traces to identify corresponding steps between said execution traces; and

generalizing said aligned execution traces to generate said reusable executable procedure.

29. The program storage device of claim 26, further comprising instructions for performing the step of augmenting said reusable executable procedure using an execution trace that is obtained during execution of said reusable executable procedure.

30. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for executing a reusable procedure, the method steps comprising:

launching a reusable executable procedure;

automatically executing procedure steps associated with said reusable executable procedure; and

relinquishing control of execution of said reusable procedure to a user, when a next step of said reusable executable procedure cannot be successfully executed.

31. The program storage device of claim 30, further comprising instructions for performing the steps of:

monitoring user actions while the user is executing steps of said reusable executable procedure;

aligning said monitored user actions to said reusable procedure; and

continuing automatic execution of said reusable procedure, if said monitored steps align with said reusable procedure.

32. The program storage device of claim 30, wherein the instructions for automatically executing procedure steps associated with said reusable executable procedure, comprise instructions for performing the steps of:

describing a procedure step to the user; and

automatically executing said procedure step, if execution of said procedure step is authorized by said user.

33. The program storage device of claim 30, wherein the instructions for automatically executing procedure steps associated with said reusable executable procedure, comprise instructions for performing the steps of:

describing a plurality of procedure steps for selection by the user; and

automatically executing a procedure step selected by the user.

34. The program storage device of claim 30, further comprising instructions for performing the step of relinquishing control of execution of said reusable procedure to a user upon request of said user.

**9. EVIDENCE APPENDIX**

(None)

**10.    RELATED PROCEEDINGS APPENDIX**

(None)